UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/021,016	12/19/2001	Alexandre Drobychev	80168-0239	8774

32658 7590 04/15/2008

HOGAN & HARTSON LLP
ONE TABOR CENTER, SUITE 1500
1200 SEVENTEEN ST.
DENVER, CO 80202

EXAMINER	
VO, TED T	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
04/15/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/021,016
Filing Date: December 19, 2001
Appellant(s): DROBYCHEV ET AL.

Michael C. Martensen
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 01/14/2008 appealing from the Office action mailed 04/19/2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Netscape Application Builder, "User Guide", Netscape Communication Corp., (1999), pages: i-xii, c1: 1-9, c2: 1-7, c5: 1-17, c6: 1-31, c7: 1-17, c8: 1-29, c10: 1-16.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

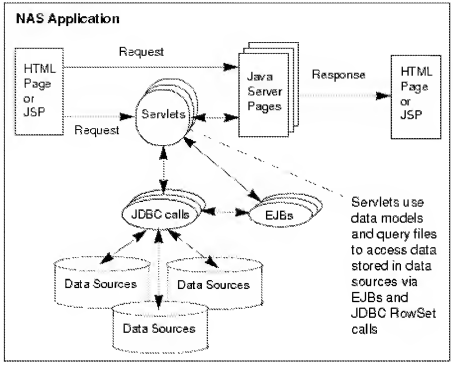
- Claims 1-16, 18-26, and 29-32 are rejected under 35 U.S.C. 102(b) as being anticipated by Netscape Application Builder (hereinafter: NAB), "User Guide", 1999.

Appellants' argued two features in Claim 1, and the two features appear the same as in independent Claim 10 and independent Claim 21. Appellants do not appeal dependents claims. Accordingly, all claims are in a group, and they stand or fall together with Claim 1.

The Claimed rejection is rearranged based on the Appeal grouping by Appellants.

As per claim 1: NAB discloses

A computer system run-time platform for providing features and services for commerce software applications, and operatively adaptable to any server platform capable of server-side presentation logic, the commerce applications platform comprising (See the figure in p C2-2),



Regarding limitation, ***a software portion configured to provide access to, and caching of, data elements*** (e.g. C2-2:"Data Sources")***, including a data and object repository, independent of the run-time platform for providing commerce software applications***;

NAB describes a builder that generates an Application Server Environment as layers. It includes a "Data Access Layer" (P. c2-2), built with:

Data access logic: This logic is built within objects for performing query files, access data stored in data sources. According to the Figure in p. C2-2 (see Figure above), it includes EJBs and JDBC calls incorporated with Servlets. Building on this logic is for access data and caching. Thus, the logic for software layer using Servlets and EJB and JDBC calls is used for being: *configured to provide access to, and caching of, data elements, including a data and object repository* as it is seen in p. C2-2, the above Figure and Data Access Logic.

The data access logic built in this layer allows to determine the type of backend data sources (see Figure in p. C2-2). The NAB provides this layer with "prebuilt extensions" and/or custom extension (See "Data Access Logic" in c2-2). It should be noted that prebuilt extensions and/or custom extensions have the ability to act as "plug-into". The extensions allow an application to communicate with cross platforms such as CICS/IMS, IBM MQ series, BEA Tuxedo Thus NAB. Thus, this logic, as the extensions, discloses *configured to provide access to, and caching of, data elements* ('data sources'), *including a data and object repository, independent of the run-time platform for providing commerce software applications* because it is built as the extensions, The extensions are independently for allowing to communicate with various run-time platforms such as CICS/IMS, IBM MQ series, BEA Tuxedo, etc.

Regarding limitation,

a software portion configured to inherit hierarchical application logic from the commerce applications platform; (See in c2-2, "servlets". The servants handle the access. See Figure in C2-2, the servlets invokes JSP and EJBs and JDBC calls);

Regarding limitation,

a software configured to provide static and dynamic presentation data for presentation by any server capable of server-side presentation logic (See the table in C2-1: Presentation layer: it uses Java servlets for the presentation logic; JavaServer Pages ('*dynamic presentation data*') and HTML files ('*static presentation data*') for the presentation layout. See Figure in C2-2 "Reponse ---> HTML page or JSP". See about "About Presentation Logic" and "About Presentation Layout", it shows the HTML and JSP are presentation data, where with JSP is as

dynamic page. See three bullets and table in C2-4, HTML and JavaScript ('scriptlet'), Enterprise JavaBeans and Servlets).

(Note : In the Appellants' summary [Spec: [28]], Appellants identified the elements as Screen templates using JSP, or HTML using scriptlets to generate dynamic data through the Servlets. Compared to the NAB: see three bullets in C2-4, NAB shows HTML pages and JavaServer Pages contain the web content, where these pages is in response from a request (i.e. HTML Page or JSP in the Figure above). See C2-3: The present logic is processed by servlets on the Netscape Application Server (i.e. "*a server capable of server-side presentation logic*"). However, see in p. C2-2, 'Data access logic", a servlet is as an extension, and allows to communicated with any type of platforms such as CICS/IMS, IBM MQ Series, etc. (i.e. "*any server capable of server-side presentation logic*").

Regarding limitation,

a software portion configured to maintain permanent and session application data persistent across user request boundaries during a single user session (See c7-11-12, it describes the term "session", the a management will be provided to trace the session. It includes using appinfo.ntv file. Furthermore, in C7-2, a set of NTV files used in servlet configuration, particularly, for session management, for maintaining and tracing data during the session. Incorporating with EJBs: see c9-2 "Entity EJBs": NAB describes Entity EJBs represents as persistent data (such as databases or document) or See c9-3, "Creating Entity EJBs", NAB describes Entity Beans represents as persistent objects, About Enterprise JavaBeans, Entity Beans, where within a servlet, these data element are across via the boundaries as the user requests a session shown in the Figure in p. c2-2);

Regarding limitation,

a software portion configured to enable access to a business object during the user session (See C2-1: "Enterprise Java Beans", the beans are built in a Business layer. See c2-2, Figure, HTML page or JSP (i.e. 'user session') request the servlets to access data source or EJBs; Thus, with a request from the user in a session, the servlets uses data models to access to the EJBs (i.e. "business object").

As per claims 2-9: The dependent Claims are not appealed. The rejection of the claims stands or falls together with Claim 1 above.

As per claim 10: The rejection of the claim 10 stands or falls together with Claim 1 above.

As per claims 11-16, 18-20: The dependent Claims are not appealed. The rejection of the claims stands or falls together with Claim 1 above.

As per claim 21: The rejection of the claim 21 stands or falls together with Claim 1 above.

As per claims 22-26, 29-32: The dependent Claims are not appealed. The rejection of the claims stands or falls together with Claim 1 above.

(10) Response to Argument

Claims 1-16, 18-26, and 29-32 are rejected and Appellants expressed an appeal on Claims 1, 10, and 21.

A. Appellants' arguments to the feature "a software portion configured to provide access to and caching of, data elements, including a data and object repository, independent of the run-time platform for providing commerce software application, appeared in Claim 1, 10, and 21.

Appellants alleged NAB does not disclose an applications platform that provides access to a data repository independent of the run-time platform. It appears the argument is that NAB does not possess the ability to plug into different backend data repositories using an abstract data layer with the application software running on top, i.e. independent of the run-time. (Brief: p. 12)

Examiner response:

Examiner disagrees. The whole argument in the Brief p. 12 and p. 13 for the above claimed limitation does not address any specific teaching in the NAB reference, as required by 37 CFR 1.111 (c), and required by 37 CFR 41.37(c)(vii), but rather mere assertion that Examiner has not shown the claimed limitation in NAB.

Yes, NAB describes a builder for EJBS and Servlets, but the NAB's teaching discloses all the features of the claims. Particularly, the teaching is for building the EJBs and servlets, which can be as extensions (C2-2: in section "Data Access Layer) built as Application Layer (Table in p. C2-1) on top of any application, legacy software ("data Source").

See C2-2:

> **Data access logic** determines what types of back-end data sources (if any) the application accesses, such as a relational database or a legacy system. In this guide, it is assumed that a data source already exists for use with your application, and that there is someone at your site, such as a database administrator, who manages it. Netscape Application Builder enables you to access your data sources via data models and data access query files.
> To legacy systems, your code may need to call into an extension. Netscape Application Server includes prebuilt extensions. These extensions allow an application to communicate with transaction-processing systems such as CICS/IMS, IBM MQ Series, and BEA Tuxedo. Additional prebuilt extensions allow communication with ERP systems such as SAP R/3. If you need to create a custom extension, use Netscape Extension Builder. (emphasis added)

The Appellants' specification [p. 8, [24]] merely assert "running independent platform" as being caused of the application layer 240 sitting on top of various of applications. It appears that the specification has no detail description for "running independent platform". However, its Application Layer "240" is not different from NAB's Application Layer which is implemented with EJBs and Servlets, as "extensions". Because these EJBs and Servlets are built as extensions (i.e. 'on top'), they have ability to plug into, and they allow an application to communicate (i.e. 'running on top') with transaction-processing systems such as CICS/IMS, IBM MQ Series, and BEA Tuxedo, where these are the different platforms and each runs independently from Servlets and EJBs (i.e. 'independent of run-time system platform for providing commerce software applications').

B. Appellants' arguments to the feature "a software portion configured to provide static and dynamic presentation data for presentation by any server capable of server-side presentation logic, appeared in Claim 1, and 10.

It appears that Appellants relied on the term "server-side presentation logic", and Appellants argued that the "presentation logic" of NAS is specific to NAS, and NAS does not supports server-side presentation logic, and NAB layer deployable on servers that support only EJBs.

Examiner response:

Examiner disagrees. It is clearly that Appellants fail to address any presentation layer and presentation logic that are built within Servlets and EJBs, where within Presentation Layer, it provides JavaServer Pages ('*dynamic presentation data*') and HTML files ('*static presentation data*') for the presentation layout (Figure in C2-2 "Reponse ---> HTML page or JSP". See about "About Presentation Logic" and "About Presentation Layout", it shows the HTML and JSP are presentation data, where with JSP is as dynamic page. See three bullets and table in C2-4, HTML and JavaScript ('scriptlet'), Enterprise JavaBeans and Servlets). As shown in page e C2-3: The presentation logic is processed by servlets on the Netscape Application Server (i.e. "*a server capable of server-side presentation logic*"). However, in p. C2-2, within 'Data access logic", a servlet is as an extension, and allows to communicated with any type of platforms such as CICS/IMS, IBM MQ Series, etc. (i.e. "*any server capable of server-side presentation logic*").

Appellants' specification has no detail description for how the application layout to support the server-side presentation logic, even they pointed out in pages 10-11. The description in the specification merely includes the static and dynamic presentation as HTML code and

Scriptlets which are generate data for presentation through the Java servlets, i.e. it is the same manner with NAB' presentation logic description. As noted by the Examiner, HTML Page or JSP shown in the Figure in C2-2 (See above Figure) are provided with static and dynamic presentation data. As noted in the section Data Access Layer, The Servlets and EJBs act only as the extensions to provide presentations in response to a request. The extensions communicate with various run-time platforms (such as CICS/IMS, IBM MQ Series, etc., i.e. "*any server capable of server-side presentation logic*"). It should be noted that NAB reference directs to a user's guide for the NAB builders. It is customary to present it as to the Netscape Application Server, but with communication various type of run-time platforms, the assertion in Appellants' argument for that NAB is specific for NAB' servers does not means the presentation logic for Servlets and EJBs built as extensions cannot not support other servers. It appears the argument does not make a connection to the recited limitation, but mere assertion. Thus, it fails to address under 37 CFR 1.111 (b)(c), fail to meet the requirement under 37 CFR 41.37(c) (vii).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

TTV

/Ted T. Vo/
Primary Examiner, Art Unit 2191
Conferees:

Wei Zhen, SPE

/Wei Zhen/
Supervisory Patent Examiner, Art Unit 2191

Tuan Q. Dam, SPE
/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192